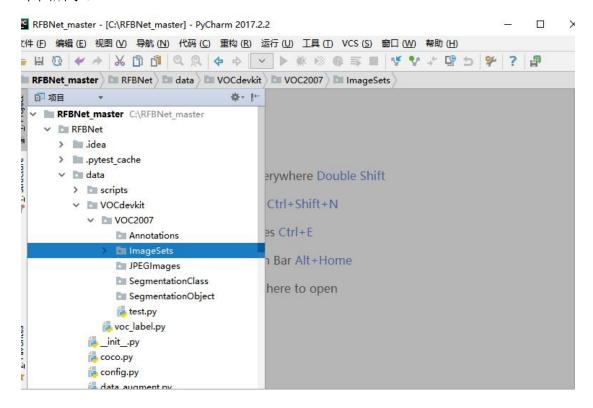
## RFBNet 模型训练教程

环境 pytorch1.0,opencv3.4,cuda9.0,cudnn7.4 其他环境为于 Anconda3.6 匹配。数据 VOC2007 模式

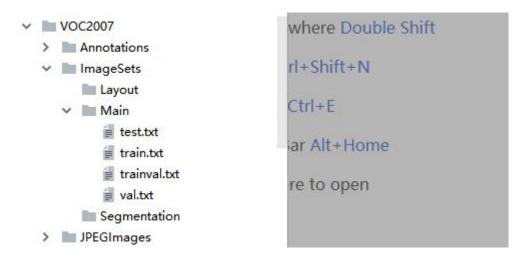
## 一、制作 VOC 数据集

使用 Labelimg 制作训练数据集文件,如果在学习和使用过程中没有该工具可以联系我。

二、将要训练的数据集的所有 xml 放在该工程目录的 data 文件下面的 VOCdevkit/VOC2007 的 Annitations 目录下,jpg 格式图片为 JPEGImages 目录下如下图所示:



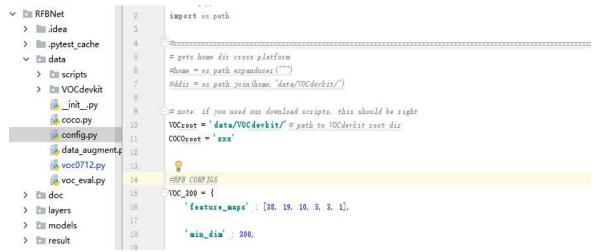
三、使用 test.py 脚本生成 ImageSets 目录下 Main 文件夹里面的训练集文件



四、修改 voc0712.py 脚本的 27 行在里面写上自己要训练的类别标签,我只有一类飞机训练数据如下图:

```
✓ ■ RFBNet_master C:\RFBN( 20
                               if sys.version_info[0] = 2:
  ∨ 🛅 RFBNet
                                 import xml etree cElementTree as ET
    > 🗎 .idea
    > ____.pytest_cache 23
                                 import xml. etree. ElementTree as ET
                       24
    V 🛅 data
      > 🛅 scripts
                           VOC_CLASSES = ('__background__', # always index 0
                       26
       > 🛅 VOCdevkit
         🍒 _init_.py
                                 'aircraft'
                       28
         🥫 coco.py
                       29
         💪 config.py
                       30
                               # for making bounding boxes pretty
         % voc0712.py 32
                                      (0, 255, 255, 128), (255, 0, 255, 128), (255, 255, 0, 128))
         👼 voc_eval.py
                       33
                        34
    > 🛅 doc
                             class VOCSegmentation(data.Dataset):
    > layers
                      36
    > models
                                  """VOC Segmentation Dataset Object
    > mesult
                                  input and target are both images
```

五、继续修改 data 目录下的 config.py 脚本里面的第 10 行为自己的数据目录如下图:



六、下载初始化训练权重到自己的的权重文件夹 weights 目录下,下载方法在我

这个工程中已经生成的文件夹如果没有可以参考原作者的方法如下: mkdir weightscd weights wget https://s3.amazonaws.com/amdegroot-models/vgg16\_reducedfc.pth adata augment.py 25 № voc0712.py 26 woc eval.py 27 > 🛅 doc 28 29 > layers > models 31 > mresult > 🛅 utils 34 🎳 vgg16 reducedfc.pth gitignore 36 **LICENSE** 37

得到一个 78.1M 的如上图权重文件

make.sh

test.sh

README.md

test RFB.py

train RFB.py

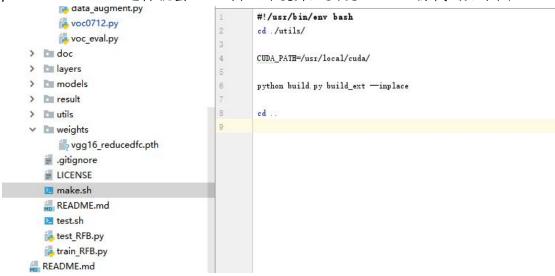
七、使用 linux 的 Terminal 或者 window 的 powershell 在该 shell 环境下 运行 sh /bash make.sh(这样就会 build 自己环境并且实现 install)源代码如下图:

39 40

41

42

43



八、以上步骤实现的训练前期的数据准备和环境搭建,接下来修改解析文件和开始模型训练,我对需要修改按照自己情况训练修改的代码片段使用 XXX 隔开方便工程化和研究时候相关代码的注解和修改醒目。

模型训练如下图:

```
✓ ■ RFBNet_master C:\RFBNet_master

Master C:\RFBNet_master
                                                                                                                                                                                                                                                                 t('-v', '-version', default='RFB_vgg',
help='RFB_vgg ,RFB_B_vgg or RFB_mobile version.')
                                                                                                                                                                                                   parser. add_argument (" -v

✓ Image: Value of the Valu
                   > 🗎 .idea
                                                                                                                                                                                                  parser.add_argument('-s', '--size', default='512',
help='300 or 512 input size.')
                           pytest cache
                    > 🛅 data
                                                                                                                                                                                                  > 🛅 doc
                    > 🛅 layers
                   > models
                                                                                                                                                                                                   parser add_argument(
'-bassnet', defaulte'./weights/vgg16_reducedfc.pth', help='pretrained base model')
parser add_argument('-jaccard_threshold', default=0.5,
type=float, help='Win Jaccard index for matching')
                    > D utils
                      weights
                                                                                                                                                                                                    parser add_argument('-b', '-batch_size', default=8,
                                       sygg16_reducedfc.pth
                                                                                                                                                                                                  parser add_argument(""h', '"-batch_size', default=0,
type=int, help="Batch size for training')
parser add_argument("—num_workers', default=0,
type=int, help="Bumber of workers used in dataloading')
parser add_argument("—nud", default=Ture,
type=bool, help="Use cuda to train model')
                               gitignore
                               LICENSE
                             make.sh
                             README.md
                             test.sh
                                                                                                                                                                                                    parser add_argument('--ngpu', default=1, type=int, help='gpus')
                               test_RFB.py
                                                                                                                                                                                                  parser.add_argument('-lr', '-learning-rate',
default=le-2, type=float, help='initial learning rate')
                                train RFB.pv
                    README.md
> 川川外部库
                                                                                                                                                                                                    parser.add_argument(' --momentum', default=0.9, type=float, help='momentum')
                                                                                                                                                                                                    parser.add_argument(
                                                                                                                                                                                                  '-resume_net', default=Wene, help='resume net for retraining')
parser.add_argument('-resume_epoch', default=0,
type=int, help='resume iter for retraining')
```

## 九、模型测试

```
▼ ■ RFBNet_master C:\RFBNet_master

✓ ■ RFBNet

                                                              parser = argparse. ArgumentParser(description='Receptive Field Block Net')
      > 🗎 .idea
                                                    21
22
23
24
25
26
27
28
29
                                                              parser add_argument (' -v', ' --version', default=' RFB_vgg'
      > _____.pytest_cache
                                                                                  help='RFB_vgg , RFB_E_vgg or RFB_mobile version.')
      > 🛅 data
                                                              parser.add_argument('-s', '--size', default='512',
help='300 or 512 input size.')
      > 🛅 layers
      > models
                                                              parser add_argument('-d', '-dataset', default='VOC'
      > 🛅 result
      > 🖿 utils
                                                                                  help='VOC or COCO version')

∨ ■ weights

             sygg16_reducedfc.pth
                                                              parser.add_argument(' -m', '--trained_model', default='weights/RFB_vgg_V0C_epoches_2990.pth'
          gitignore .gitignore
                                                                                  type=str, help='Trained state_dict file path to open')
          LICENSE
                                                              make.sh
         README.md
                                                              parser.add_argument(' -- cuda', default=True, type=bool,
help='Use cuda to train model')
         test.sh
                                                              parser add_argument ("repu', default=False, type=bool,
help='Dse cpu mms')
parser add_argument ("-retest', default=False, type=bool,
help='test cache results')
         train RFB.pv
      README.md
> ||||| 外部库
```

九、温馨提示:为了代码执行和训练更加顺利进行,训练和测试最好都写成 shell 在终端执行和测试,这个模型 IDE 里面我在自己环境下使用发现训练可以测试报错,直接 shell 执行 ok。

20181227 安芯

感谢原作者开源,在此祝你好运训练不出 bug!